AD-761 494

MAINTENANCE SCHEDULING LIMITATIONS

Joseph E. Boyett, Jr.

Air Force Institute of Technology
Wright-Patterson Air Force Base, Ohio

24 April 1973

DISTRIBUTED BY:

NTIS
National Technical Information Service
U. S. DEPARTMENT OF COMMERCE
5285 Port Royal Road, Springfield Va. 22151

UNITED STATES AIR FORCE

AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

24

MAINTENANCE SCHEDULING LIMITATIONS

Joseph E. Boyett, Jr., Major, USAF

AU-AFIT-SL-2-73

AFIT/AU TR

UNCLASSIFIED

Security Classification

DOCUMENT CONTROL DATA - R & D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY (Corporate author) AFIT/SLGM	2a. REPORT SECURITY CLASSIFICATION Unclassified 2b. GROUP
3. REPORT TITLE MAINTENANCE SCHEDULING LIMITATIONS	
4. DESCRIPTIVE NOTES (Type of report and inclusive dates) AFIT/AU TR	
5. AUTHOR(S) (First name, middle initial, last name) Joseph E. Boyett, Jr.	
6. REPORT DATE 24 April 1973	7a. TOTAL NO. OF PAGES ~~22~~ 24 7b. NO. OF REFS 0
8a. CONTRACT OR GRANT NO b. PROJECT NO c. d.	9a. ORIGINATOR'S REPORT NUMBER(S) AU Tech Report AU-AFIT-SL-2-73 9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report) None
10. DISTRIBUTION STATEMENT Approved for public release; distribution unlimited.	
11. SUPPLEMENTARY NOTES Approved for Public Release; IAW AFR 190-17 JERRY C. HIX, Captain, USAF Director of Information	12. SPONSORING MILITARY ACTIVITY School of Systems and Logistics Wright-Patterson AFB, Ohio 45433

13. ABSTRACT

This paper discusses maintenance scheduling problems which must be resolved during phase II of the STALOG development. Performance measures associated with scheduling are identified as being extremely critical since they must be clearly established prior to developing specific scheduling techniques. A brief review of sequencing techniques allows the reader to visualize how different heuristics might be used in a dispatching model developed for specific applications. Recommendations are tentative since very little research has been conducted on complex problems of this type.

DD FORM 1 NOV 65 1473

UNCLAS

Security Classification

Security Classification

14 KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
Maintenance						
Objectives						
Scheduling						
Dispatching						
Sequencing						

Security Classification

MAINTENANCE SCHEDULING LIMITATIONS

A School of Systems and Logistics Technical Report

Air University

Air Force Institute of Technology

Wright-Patterson AFB, Ohio

By

Joseph E. Boyett, Jr.
Major, USAF

April 1973

Approved for public release;
distribution unlimited

TABLE OF CONTENTS

Introduction

Scheduling problems have plagued managers since time immemorial; however, only within the last two decades have advances been made in scheduling theory which offer some hope for reducing the vagaries of this difficult task. In particular, the Air Force is deeply concerned about how schedules are prepared and their effects on mission capability. This report addresses performance measures, sequencing techniques and a simple dispatch model. It does not attempt to cover all areas of scheduling theory nor does it delve into the selected areas in great depth. Rather it points out specific areas that must be considered during the development of a program which seeks to use computers as aids in making scheduling decisions in an environment that is very uncertain. The following discussion addresses scheduling as a factor to be incorporated in the overall STALOG conceptual model.

Performance Measures

The concept of assigning tail numbers to a specific mission far enough in advance to prepare a weekly or monthly schedule has an adverse effect on in-commission rates and indirectly on the response capability of the aircraft fleet. Response capability as used here refers to the ability of an aircraft to respond to a mission and is related to operational ready time. Flying hour capability is a more common term and refers to the potential number of flying hours that a fleet of aircraft can produce subject to fixed resource quantities. If assumptions are made that one is not concerned about the other factors associated with a mission, e.g. combat crew, and if the fleet is homogeneous, then maximum response capability is reached when the in-commission rate is maximized. These assumptions, of course, are not totally logical; however, mission departures can normally be permitted to vary over some time interval without producing significant impact on the overall Air Force mission. Equally important, the degree of homogeneity is very high for most aircraft types assigned to a base. As a result, substitution of aircraft for a particular mission may not be a major problem. However, if mission departure times are of overriding importance, then one must view this objective as being costly with tradeoffs between increasing the number of aircraft or increasing resource quantities. Succinctly stated, for a given set of resources, response capability is a positive function of in-commission time and is inversely related to the rigidity of mission schedules.

One of the major difficulties encountered when discussing response capability with managers is the almost universal belief that one can not increase response capability without either increasing resource quantities or increasing utilization of existing resources. Utilization in particular is a favorite target but sequencing (scheduling) theory is rarely mentioned. The following simple example shows how in-commission time, i.e. response capability, can be significantly affected by the sequencing rule used. Assume that at 0100 we have two aircraft, tail numbers one and two, each out of commission for separate malfunctions. Each malfunction requires exactly one manhour for repair which may be performed by one man in one hour or two men in one-half hour. Given that we have two men available from 0100 to 0200, we then are faced with the task of deciding how to allocate their time. First we may consider assigning a mechanic to each aircraft with the result that both aircraft are repaired by 0200. Using this method, we find that over the period from 0100 to 0200 the mechanics are utilized 100% of the time and that the aircraft are in-commission zero percent of the time.

Perhaps we may wish to try some other method and hit upon assigning both mechanics to aircraft number one first, and upon completion of it, then assigning them to aircraft number two. Under this method the mechanics are still utilized 100% of the time from 0100 to 0200, but we detect a significant change in aircraft in-commission rates. Aircraft number one was undergoing repair from 0100 to 0130 and in-commission from 0130 to 0200. Aircraft number two was awaiting repair from 0100 to 0130 and undergoing repair from 0130 to 0200. The aircraft in-commission rate using this schedule is 25% over the period from 0100 to 0200. One should not be deceived by the simplicity of the foregoing problem and misled into thinking that these results can not be shown in the real world. They can be obtained although the actual change in in-commission rates may not be so dramatic. The example clearly illustrates that we can change in-commission rates without changing resource utilization rates or resource quantities, merely by selecting a particular sequencing rule. Since in-commission time translates into ability to respond, we have shown that the fleet's ability to respond to unknown requirements is increased.

Other scheduling objectives may be useful if due dates are of overriding importance. Due dates in this context refer to the time that an assembly must be completed and is analogous to mission departure times for aircraft. For example, one may wish to minimize the mean number of missions that are delayed beyond a scheduled departure time. Another objective that may be of value is to minimize the mean tardy time or rather to minimize the mean time that missions are delayed beyond a scheduled departure time. Although both objectives are frequently used, the resource allocation techniques that are most useful for maximizing in-commission time frequently produce poor schedules when the scheduling objective is due date oriented.

Based on the above discussion, one can not state a priori that a particular resource allocation technique is useful until a specific objective has been clearly stated. For this reason, a thorough analysis of a particular unit's mission must be made and a single clear, simple scheduling objective must be adopted as policy. The particular constraints that are binding on a unit must be identified and incorporated into the scheduling technique. Examples of such constraints are 1) to keep each aircraft flying at approximately the fleet utilization rate or 2) missions that require a particular aircraft tail number.

The emphasis of the above paragraphs has been toward a single scheduling objective subject to specific constraints. This particular point has to be fully recognized because progress toward developing an algorithm to take over some of the routine scheduling steps and to assist in the more critical scheduling decisions is doomed unless a single objective is stated and constraints listed. This must become the first order of business for any project to computerize scheduling.

This paper has not provided an answer for a single scheduling objective, however, prime candidates are 1) maximize in-commission time and 2) minimize mean number of delayed departures.

Sequencing Techniques

The resource allocation problem that exists in aircraft maintenance, and similar maintenance functions, is stochastic and possesses numerous uncertain elements. Modern maintenance managers, however, are closely tied to the past and want detailed monthly and weekly schedules that indicate start and end times for specific events. As stressed earlier, such a rigid schedule can be followed without deviation only if either response capability is reduced or resource quantities are increased. Neither of these alternatives are appealing in an economy which demands that the military maximize the output of every budget dollar. In combination, these system characteristics and broad objectives eliminate the more common mathematical programming techniques used to prepare schedules. The remaining discussion uses assemblies and jobs in a very general manner and deviates somewhat from accepted usage. In this context, however, assembly is analogous to the end item; e.g. aircraft, and job is analogous to a typical task on the end item.

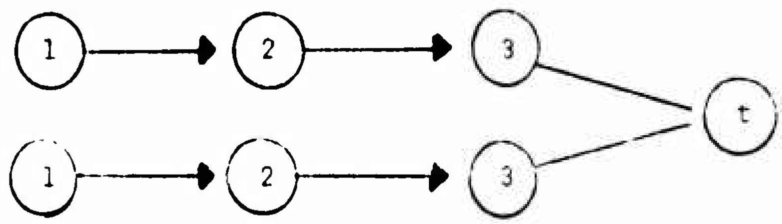
The most promising technique of resource allocation in a stochastic system appears to be dispatching. Dispatching contrasts sharply with scheduling since a decision to allocate a particular resource subset is made each time the maintenance system changes state. In this context, the system changes state when either a new job is imposed on the system or an old job is completed. Scheduling in a stochastic environment on the other hand looks forward over time and strives to achieve some objective by predicting what will happen. Unfortunately, in a stochastic system, schedules are made to be broken. Recognizing this, one may wish to revise the schedule each time the system changes state; however, this is equivalent to dispatching.

Both techniques have disadvantages, however, and an alternative might be to partition the maintenance workload into two subsets, one of which is composed of assemblies with small job standard variances and a relatively large return associated with completing the jobs at a predetermined time. The other subset would be composed of assemblies with large job standard variances and minimal return associated with completing the jobs at a predetermined time. Examples of jobs which may qualify assemblies for the first subset, subset A, would include preflight, postflight, scheduled inspections and munitions loading. The second subset, subset B, would be composed of assemblies with jobs such as unscheduled maintenance and in shop work.

The first subset of assemblies, subset A, permits scheduling of events prior to their occurrence, however, even the jobs of these assemblies will have some variation in process time which will cause slippage of their successor's start times. This problem may be minimized by maintaining spare resources specifically for these jobs, subset A, or by permitting jobs associated with subset A to preempt resources from jobs associated with subset B. Either of these methods of reducing slippage of subset A's jobs may or may not be feasible, however, an a priori statement concerning feasibility can not be made at this time because of the lack of data.

Assuming that it is desirable to define a subset of assemblies which should be scheduled as stated above, then the most attractive technique for allocating resources to those jobs associated with the second subset is by dispatching with options for maintenance control to override if necessary. There are numerous heuristics (dispatch rules, priority rules, rules of thumb) which may be used individually or in combination and consistently allocate resources such that good results are obtained for regular measures of performance. The regular measures of performance useful in a stochastic system are mean flow time, late time, and tardy time. Considerable research has been conducted by simulating simple dynamic job shops with both flow routing and random routing to establish precedence between operations. Shortest processing time rules have consistently minimized mean flow time for these simple shops.

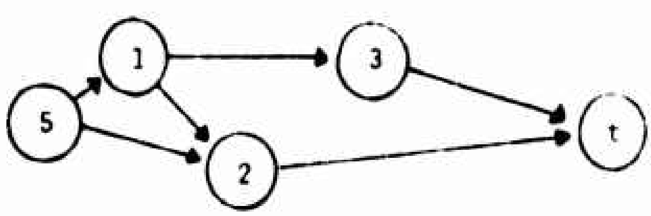
Very little research has been conducted on assembly shops however. For the very simplest assembly shop an example of which is illustrated in Figure 1 with the sink node (t) as a dummy assembly operation, the shortest process time rule continued to produce good results.



ASSEMBLY SHOP WITH FLOW ROUTING

Figure 1

The only research published on a 20 operations complex assembly shop, an example of which is illustrated in Figure 2, concludes that such simple rules as first come first served, consistently minimized mean flow time. It remains to be seen whether or not a skilled individual can, when provided a large set of heuristics, consistently produce lower mean flow times by selectively employing them depending on which state the maintenance system may be in. Since the maintenance system can be in one of an almost infinite number of states, it may be possible to manually produce better results, at least until we know more about the relationships among system states, heuristics, and performance measures.



ASSEMBLY SHOP WITH GENERAL ROUTING

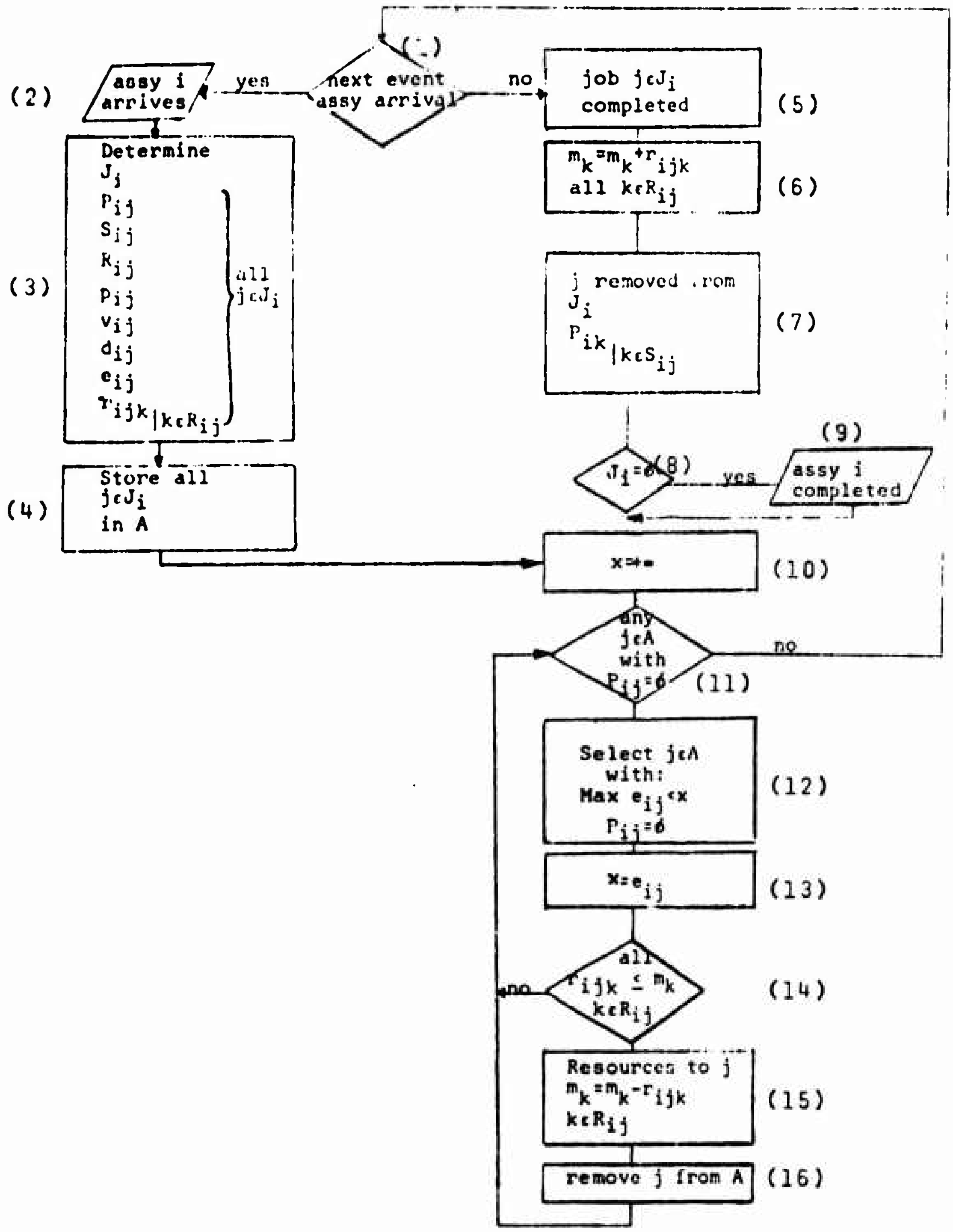
Figure 2

Heuristics may be partitioned into two sets, dynamic and static, based on their response to changes in the system. Heuristics that permit changes in job priorities are typically classified as dynamic. Examples of dynamic heuristics are 1) total number of jobs remaining on an assembly and 2) shortest remaining process time. On the other hand static heuristics assign job priorities and once assigned these priorities never change. Examples of static heuristics are 1) total number of initial work content or 2) total number of initial jobs.

The main reason for partitioning heuristics into dynamic and static sets is the a priori conclusion that dynamic heuristics, which incorporate the current state of the system, are inherently more efficient than static heuristics, which are based on historical events.

Another and more useful way of partitioning heuristics is based on the source of data. Heuristics that are based on specific information associated with jobs may be grouped together as job oriented. Examples of job oriented heuristics are 1) job process time and 2) number of predecessor jobs per job. Heuristics that are oriented toward the assembly are grouped as assembly oriented. A heuristic that is based on the total number of expected work hours per assembly is an example of an assembly oriented heuristic.

A third heuristic group based on data source uses due dates as the primary means of assigning dispatch priorities. This particular group may or may not be distinct from the job and assembly based groups. For example, a due date that is established by using the total initial work content of an assembly is necessarily related to the assembly. However some due dates are established independently of any job or assembly data. As a result, due date based heuristics might also fall into one of the earlier groups. This particular relationship between due dates and job or assembly data indicates that considerable work must be done before one adopts a particular method for establishing due dates. Nonetheless, it is generally concluded that the most effective techniques for establishing due dates should incorporate some job or assembly data.



(1)
next event
assy arrival
yes
no
(2)
assy i
arrives
(3)
Determine
J_i
P_{ij}
S_{ij}
R_{ij}
p_{ij}
v_{ij}
d_{ij}
e_{ij}
$r_{ijk}|k\epsilon R_{ij}$
all
$j\epsilon J_i$
(4)
Store all
$j\epsilon J_i$
in A
(5)
job $j\epsilon J_i$
completed
(6)
$m_k=m_k+r_{ijk}$
all $k\epsilon R_{ij}$
(7)
j removed from
J_i
$P_{ik}|k\epsilon S_{ij}$
(8)
$J_i=\phi$
yes
(9)
assy i
completed
(10)
$x=+\infty$
(11)
any
$j\epsilon A$
with
$P_{ij}=\phi$
no
(12)
Select $j\epsilon A$
with:
Max $e_{ij}<x$
$P_{ij}=\phi$
(13)
$x=e_{ij}$
(14)
all
$r_{ijk}\le m_k$
$k\epsilon R_{ij}$
no
(15)
Resources to j
$m_k=m_k-r_{ijk}$
$k\epsilon R_{ij}$
(16)
remove j from A

Figure 3

Table 1

Sets

$A = \{x | x \epsilon J_i,\ i \epsilon N,\ x \text{ is awaiting work}\}$

$J_i = \{x | x \text{ is a job of assy } i\}$

$M = \{x | x \text{ is a resource type available for dispatch}\}$

$N = \{x | x \text{ is an assy with jobs not completed}\}$

$P_{ij} = \{x | x \text{ is a predecessor of job } j \text{ on assy } i\}$

$R_{ij} = \{x | x \text{ is a resource type required by job } j \text{ of assy } i\}$

$S_{ij} = \{x | x \text{ is a successor of job } j \text{ on assy } i\}$

Scalars

d_{ij} = due date of job j on assy i

e_{ij} = dispatch priority of job j on assy i

$m_{k|k \epsilon M}$ = quantity of resource k available for dispatch

p_{ij} = process time of job j on assy i

$r_{ijk|k \epsilon R_{ij}}$ = quantity of resource k required for job j of assy i

v_{ij} = process time variance of job j on assy i

Note: All jobs on an assembly (assy) may have either equal due dates or equal dispatch priorities or both depending upon the particular dispatching procedure used.

Dispatching Model

Dispatching is a deceptively simple process since it can easily provide a feasible solution to any scheduling problem. Note the solution is merely feasible and not optimum or good. Of course, by stating an objective and constraints, we indicate that we are interested in finding the optimum schedule even though we sometimes are forced to accept a good schedule. Even though objectives change, the dispatching process does not. The following narrative identifies the key events, data elements, and decisions that are made in a typical dispatching environment as depicted in the model in figure 3 using the symbology listed in table 1. In the discussion, assembly is analogous to an aircraft and the resources include personnel, equipment, facilities and parts.

As stated earlier, only two events cause the system to change states. This condition is indicated by block 1, figure 3, which branches to either an assembly (assy) arrival or a job completed. An assy arrival, block 2, with its set of jobs imposes new demands upon the system which may or may not possess free resources to commence work on some of the new jobs. The first requirement is to obtain the necessary information for scheduling purposes as indicated by block 3.

Block 3 states that the set of jobs (J_i) on the assy is determined along with each job's predecessors (P_{ij}) and successors (S_{ij}). The predecessors of a particular job j consists of those jobs which must be completed prior to starting job j. Conversely the successors of job j consists of those jobs for which job j must be completed prior to their start time.

Block 3 also requires that the set of resource types (R_{ij}) required to complete job j be identified together with the specific quantities of each resource type (r_{ijk}). The expected process time (P_{ij}) and process time variance (v_{ij}) are obtained for each job together with each job's due date (d_{ij}) and dispatch priority (e_{ij}). The information required by block 3 is used by most dispatching systems albeit rather implicitly sometimes and frequently without due consideration for their effect on decisions.

Block 4 stores all new jobs in an awaiting work set (A). The set of jobs A consists of those jobs which are either waiting for resources or can not be allocated resources because some predecessor job has not been completed.

Block 10 sets the variable x to infinity for future use in searching sequentially through the set of jobs which are being considered for dispatching resources. Block 11 in turn poses the questions, are there any jobs with predecessors all completed; i.e. the set P_{ij} is a null set, and which are not presently in work as denoted by their presence in A?

Block 12 is the real key for any dispatching system and it is here where particular dispatching rules or combinations of them are employed. First of all, we wish to consider only those jobs whose predecessors are all completed and within this set we wish to select that job which has the highest dispatch priority (e_{ij}). Various heuristics will be employed in assigning dispatch priorities subject to whatever constraints the system may impose. This particular block implicity assumes that two jobs can not have precisely the same dispatch priority. This is equivalent to stating that rules are used to break ties of simple rules if they occur.

Given that a job has been selected for work, then the next step is to determine if resources are available. Block 14 indicates this decision by determining if the required resources (r_{ijk}) are less than the free resources (m_k). If resources are available, then they are dispatched as shown in block 15 and free resources (m_k) reduced by the number of resource units dispatched (r_{ijk}). As a final act, job j is removed from the awaiting work set A.

If at block 14 resources are not available, then the system sequences to the next available job by returning to block 11. This might also occur at block 16 after a job j has just received its resources. This cycle continues until all jobs in set A have been considered at which time the system returns to block 1 pending the next event.

If the next event is a job completion, show in block 5 as job j of assy i, then the system would move down the right side of the model. The first event would be to add all of the resources (r_{ijk}) freed by the completion of job j to the various resource groups (m_k) as shown in block 6.

Block 7 indicates that job j as a predecessor is removed from all of the predecessor sets of its successors. This information is used to provide current visibility on a particular job, awaiting work, and its candidacy for work.

Block 8 inquires whether or not all jobs on a particular assembly are completed. If so, then the assembly exits the system as shown in block 9. At this point the system goes to block 10 which is common to both distinct events.

The system as depicted is a closed loop which reacts to the stimuli provided by either an assembly arrival, block 2 or a job completion, block 5. Although numerous complexities could be introduced, they would overshadow the model's purpose which is to present the way dispatching works in practice.

Recommendations

The above discussion hopefully provides the necessary ground work for the following unranked recommendations:

1. Identify each distinct job associated with each aircraft type and CEM item.

2. For each distinct job identify:

   a. All resources required to complete the job.

   b. The expected job process time.

   c. The job process time variance.

   d. The job's direct predecessors.

3. Determine which measure of scheduling performance will be used.

4. Identify those jobs which must be scheduled daily, recognizing that the remaining jobs that exist or arrive will receive resources by dispatching.

5. Determine through simulation which heuristic(s) consistently produce good schedules relative to the measure of scheduling performance. I would recommend that the following heuristics be included in the set of heuristics tested.

   a. First arrive first served; all jobs are assigned a priority equal to the arrival time of the aircraft. Resources are dispatched to the job with lowest priority.

   b. First come first served; all jobs are assigned a priority equal to the time at which its last predecessors job is completed. Resources are dispatched to the job with lowest priority.

   c. Shortest job process time; all jobs are assigned a priority equal to the expected process time. Resources are dispatched to the job with lowest priority.

d. Shortest assembly path; all jobs are assigned a priority equal to the expected minimum flow time for the parent assembly. Resources are dispatched to the job with the lowest priority.

The above recommended heuristics will have numerous ties within a given assembly. Such ties may be broken by a number of methods using secondary heuristics; however, none are recommended here.